Chapter 2

CloudVision eXchange (CVX)

CloudVision eXchange (CVX) provides a single access point for real-time provisioning, orchestration and integration with third-party controllers. CVX aggregates and distributes operational state information across a set of EOS switches to support applications that provide network services.

Sections in this chapter include:

2.1 CVX Overview

A CVX deployment includes CVX and a set of CVX clients to which CVX provides services. CVX is not part of the data plane, nor does it receive data-path traffic. All CVX components exist as agents that run on EOS instances.

For more information, see:

- "System Requirements"
- "CVX Infrastructure" on page 5
- "CVX Features" on page 5
- "CVX Clients" on page 5

2.1.1 System Requirements

Certain hardware and software is required to be able to use CloudVision eXchange in your CloudVision virtual appliance implementation.

Note The CloudVision eXchange should be installed on a single system along with CloudVision Portal.

Table 2-1 lists the minimum hardware and software required to use CloudVision eXchange.

Table 2-1 System Requirements

Required Hardware
The hardware required to use the CloudVision eXchange are: • CPU: 4 cores (base), 8 cores (recommended) • RAM: 4G (base), 8G (recommended) • Disk: 4G
Required Software
The software required to use the CloudVision eXchange are: • EOS switches: Recommend 4.16.8M or later **Note** It is a best practice and highly recommended that the version of CVX should match the version running on the switches. • CloudVision Portal: version 2016.1 (CloudVision Portal software is required if you want to use it in conjunction with CloudVision eXchange. If you plan to use only CloudVision eXchange, CloudVision Portal software is not required.)

Note CVX does not support live vMotion. If the Hypervisor environment is set up for live vMotion, it has to be disabled for the CVX VMs.

2.1.2 CVX Infrastructure

CVX provides a single integration point into network-wide services running across CVX clients. CVX is typically deployed as an EOS instance running on a VM (vEOS). The CVX infrastructure consists of a CVX instance functioning as a server and a set of CVX clients. The CVX server uses a heartbeat keepalive (KA) mechanism to maintain contact with its clients.

When de-configuring or shutting down CVX, client services should be shut down first.

2.1.3 CVX Features

CVX manages communications among the network CVX clients, and provides an integration point for services to those clients. CVX also discovers the physical network topology by aggregating topology information it receives from its client devices.

2.1.4 CVX Clients

CVX client is the agent that allows a switch to interact with a CVX server to access CVX services. Enabling the CVX client includes providing the IP address or host name of the device running CVX. The CVX client can then access services that are enabled on the CVX server.

The CVX client must be enabled to access the CVX server and the services it offers. Individual services may require additional configuration statements.

Services should be shut down or de-configured on clients before shutting down or de-configuring CVX. CVX-controlled switch features may continue to run after shutting down CVX if they are not explicitly shut down or de-configured prior to shutting down CVX.

2.2 CVX Services

CVX services are applications that run on top of the CVX infrastructure, and are accessed by CVX clients through the CVX server. All CVX services are maintained by version level; client switches negotiate the version they use when connecting to the server. This allows multiple switches that run different EOS versions to connect to the same CVX server.

The following sections briefly describe some of the services available to CVX clients through CVX:

- "OpenStack Service"
- "VXLAN Control Service" on page 6
- "Hardware Switch Controller (HSC) Service" on page 6
- "Network Topology Service" on page 6

2.2.1 OpenStack Service

The OpenStack service on CVX allows the networking component of an OpenStack deployment (also known as Neutron) to share state with CVX.

When deployed, this integration allows CVX to send state about the logical networks created in the OpenStack cloud to the CVX clients that configure the network.

More information on OpenStack software can be found in its online documentation at http://docs.openstack.org/.

2.2.2 VXLAN Control Service

The VXLAN control service allows hardware VXLAN tunnel end points (VTEPs) to share state with each other in order to establish VXLAN tunnels without the need for a multicast control plane. Configuration is required both on the client switches and in CVX.

2.2.3 Hardware Switch Controller (HSC) Service

Traffic between virtual machines which share a physical host (or between virtual machines and the rest of the network) is forwarded by virtual switches. The management and configuration of virtual switches uses the Open vSwitch Database (OVSDB) management protocol, as described in RFC 7047.

The hardware switch controller (HSC) service provides an integration point between OVSDB controllers and the VXLAN control service, allowing exchange of state information among virtual and hardware switches.

2.2.4 Network Topology Service

The network topology service gathers information from CVX clients to provide a view of the physical topology of the network. Aggregated information gathered by the network topology service is used by other CVX services, and can be viewed on the CVX server.

2.3 Deploying CVX

CloudVision Exchange (CVX) can be deployed on KVM and ESXi. The required EOS version and Aboot version vary depending on whether you are deploying CVX on KVM or ESXi.

For the detailed steps to use to deploy CVX, see:

- "Deploying CVX on Kernel-based Virtual Machine (KVM)"
- "Deploying CVX on VMware ESXi" on page 8

2.3.1 Deploying CVX on Kernel-based Virtual Machine (KVM)

Complete the following steps to install CVX on Ubuntu/KVM. Once the installation is complete, you can begin the CVX configuration process.

Note Make sure you select versions of EOS and Aboot that meet the minimum requirements for CVX. The supported versions are:

- EOS (version 4.16.8M or later)
- Aboot-veos-serial-8.0.0.iso (located in the vEOS section of the download)

Pre-requisites

Before you begin the procedure, make sure that:

- Install **qemu-kvm**, **libvirt*,** and all related dependencies using yum (RHEL7/CentOS7) and apt-get (Ubuntu).
- Two bridges are configured for use by the KVM VM, and that you have the names of the bridges. (Steps are included in the procedure to add bridges, if they are not already configured.)

Note The bridges must be configured to persist (**brctl** commands do no persist across reboots). You can use Network Manager (or another application available to you) to complete this configuration.

- You have both **generateXmlForKvm.py** and **cvpTemplate.xml**. They are required to complete the procedure. You can find them in the CVP tarball for Ubuntu.

Complete the following steps to install CVX.

Step 1 Download the Aboot and EOS files from: https://www.arista.com/en/support/software-download/.

Step 2 Use **sudo su** to acquire superuser privileges, which are required to complete some of the installation steps.

Step 3 Confirm that KVM is running on the server by entering the following command:

```
virsh -c qemu:///system listAb
```

The command output should match this example:

```
Id Name                 State
----------------------------------
$
```

Step 4 If the output does not look correct (previous step) go to for additional assistance: https://help.ubuntu.com/community/KVM/Installation/.

Step 5 Use the following command to convert the **vmdk** file to qcow2:

```
qemu-img convert EOS_4_16_8M.vmdk -O qcow2 EOS.qcow2
```

Note Step **6** and **7** are required if you do not already have 2 bridges defined in different subnets. If the bridges exist, go directly to step **8**.

Step 6 Use **brctl** to add bridges for the KVM VM to use (br1 and br2 can be any names you choose).

```
brctl addbr br1
brctl addbr br2
```

ifconfig can be used to identify Ethernet ports to be bridged. Once you identify the ports, add them to the bridges.

Example: `brctl addif br1 enx803f5d086eae`

Step 7 Confirm that the bridges are up using **brctl show**.

- Enter: `ifconfig br1 up`
- And: `ifconfig br2 up`

Note The following step uses a number of input parameters (the number required vary depending on your server setup). To ensure the command executes successfully, we recommend that you type it into a scratch pad and edit as needed before typing it into the Linux Terminal.

Step 8 Use the following command to generate **cvx.xml**, which will be used to setup the CVX VM.

```
generateXmlForKvm.py
```

Example:

```
python generateXmlForKvm.py -n cvx --device-bridge br1 --cluster-bridge br2 -e
/usr/bin/kvm -i cvpTemplate.xml -c
/home/myname/Downloads/Aboot-veos-serial-8.0.0.iso -x
/home/myname/Downloads/EOS.qcow2 -b 8192 -p 2 -t

   -n cvx: VM name.
  --device-bridge br1: This is the name you gave the bridge - br1 or anything else.
   --cluster-bridge br2: Cluster bridge if clustering servers.
   -i cvpTemplate.xml: Path to XML file input template.
```

```
-k: VM ID number used by virsh. If not entered, a random number is assigned.
-b 8192: 8G of RAM.
-p 2: # of CPU cores.
-c: Path to Aboot file.
-x: Path to qcow2 file created in step 3.
-t: This parameter indicates the file defined by -x is for CVX.
-e '/usr/bin/kvm': Ubuntu path to KVM.
      (for RHEL KVM this is: -e 'usr/libexec/qemu-kvm')
-o: XML file used by virsh to define the KVM VM.
```

Step 9 Run the following commands:

```
virsh define cvx.xml
virsh start cvx
virsh console cvx
```

Step 10 (Optional) To configure CVX to start automatically, enter:

```
virsh autostart cvx
```

You are now ready to begin the CVX configuration (see "CVX Configuration" on page 14).

2.3.2 Deploying CVX on VMware ESXi

Complete the following steps to install CVX on ESXi. Once the installation is complete, you can begin the CVX configuration process.

Note Make sure you select versions of EOS and Aboot that meet the minimum requirements for CVX. The supported versions are:

- EOS (version 4.16.8M or later)
- Aboot-veos-8.0.0.iso (located in the vEOS section of the download)

Complete the following steps to install CVX.

Step 1 Go to:
www.arista.com.

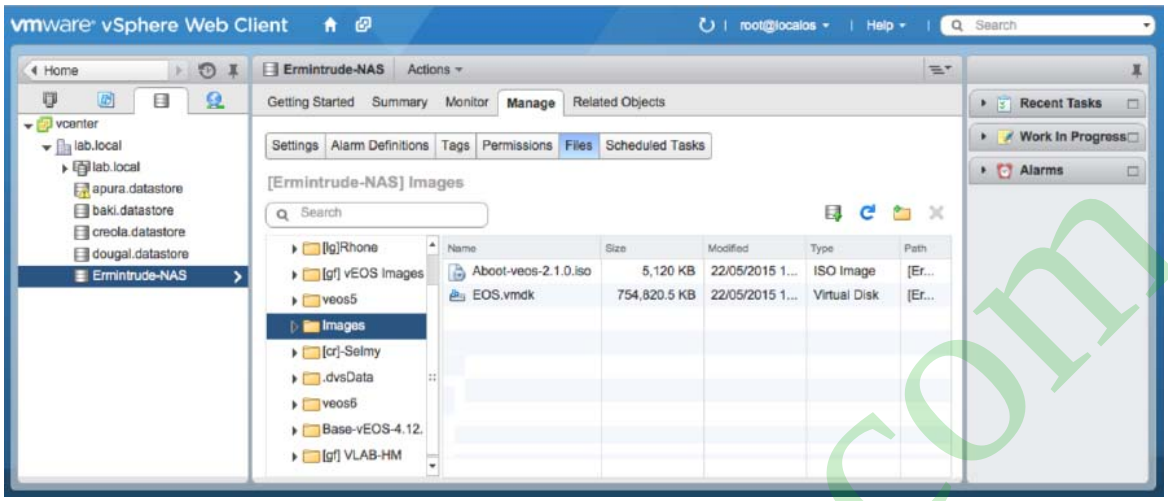
Step 2 Select **Support** > **Software Download**.

Step 3 From the software download page, expand **vEOS,** and then download **Aboot-veos-8.0.0.iso**.

Step 4 From the software download page, expand **Active Releases** > **4.16** > **EOS--4.16.8M** to download **EOS--4.16.8M.vmdk**.

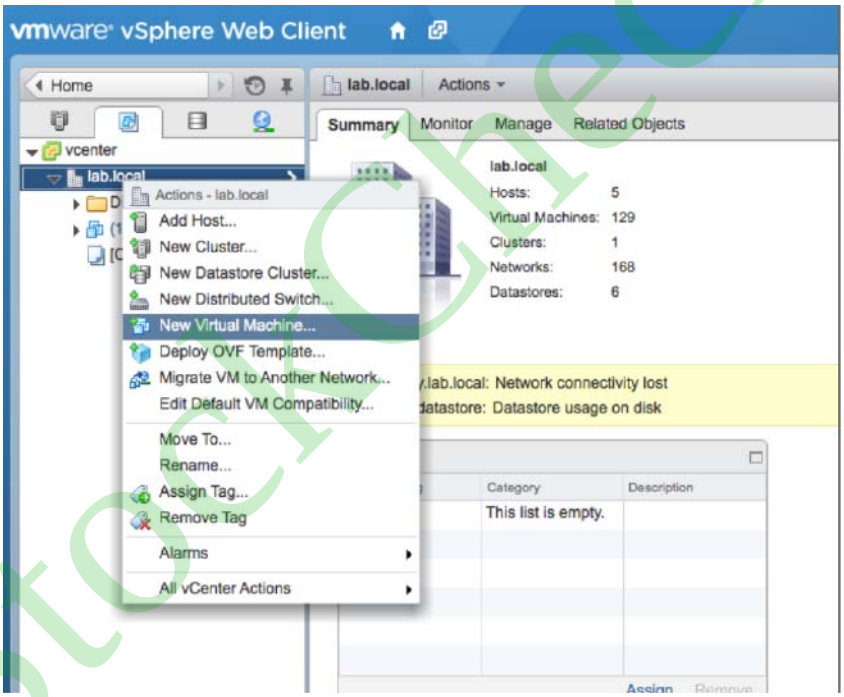
Step 5 Load the files you downloaded into a filestore location within the VMware vSphere environment (Figure 2-1 on page 9).

Figure 2-1: Loading the files into the VMware vSphere environment



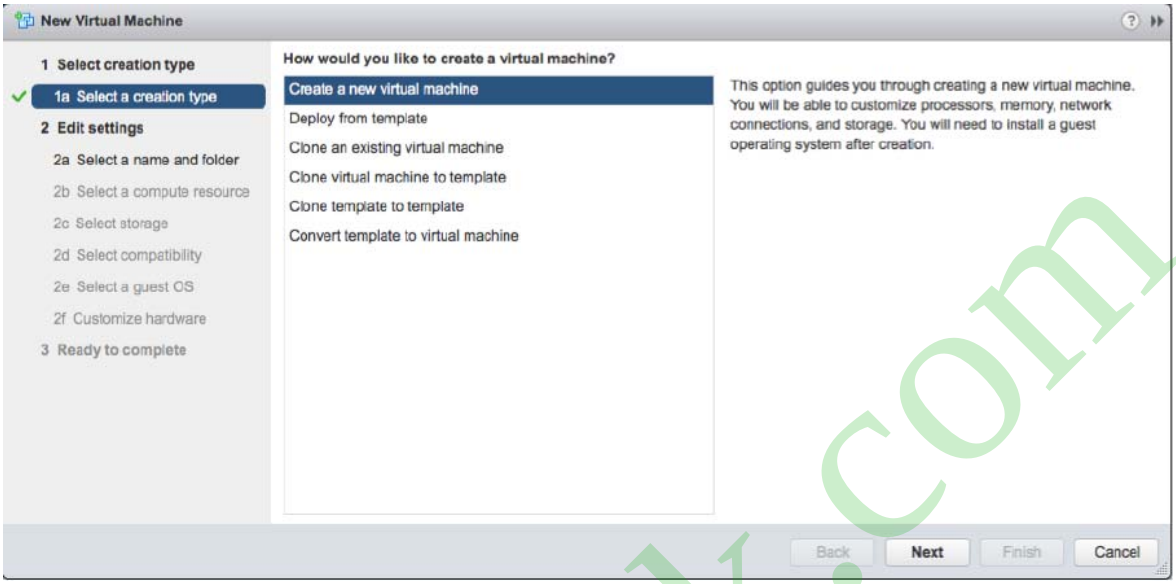
Step 6 Right-click the **filestore location** you selected, and choose **New Virtual Machine** (Figure 2-2).

Figure 2-2: Selecting New Virtual Machine



The New Virtual Machine dialog appears (Figure 2-3 on page 10).

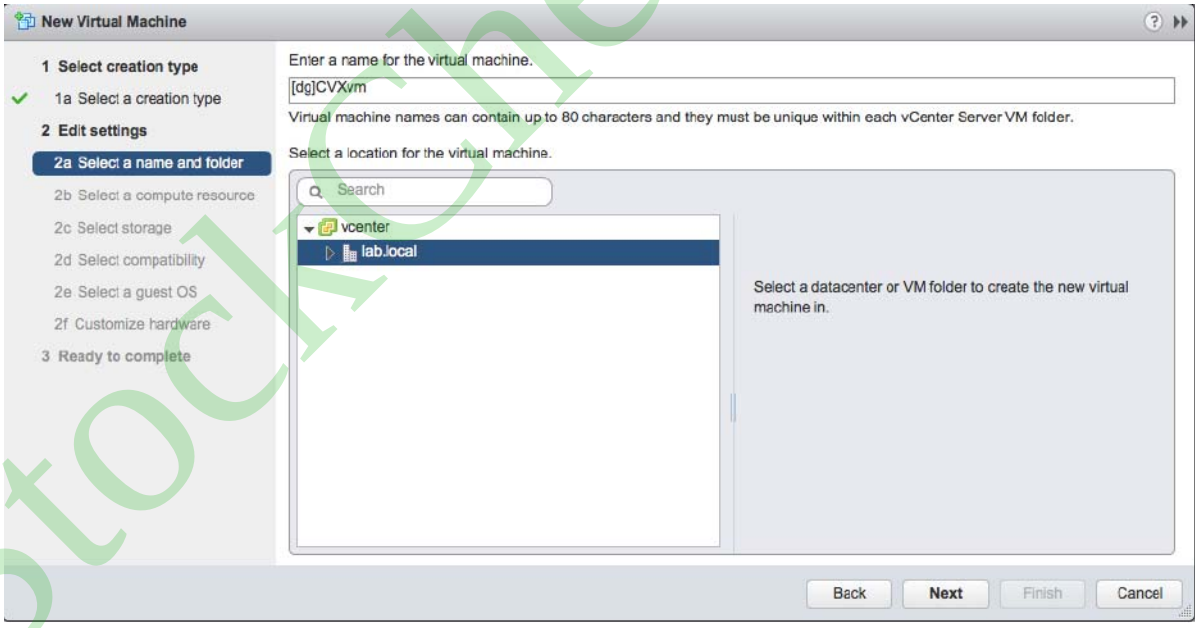
Figure 2-3: New Virtual Machine dialog



Step 7 In the New Virtual Machine dialog, select **Create a new virtual machine**, and then click **Next**.

The dialog refreshes, showing options for the new Virtual Machine (Figure 2-4).

Figure 2-4: New Virtual Machine dialog (naming and selecting the location)

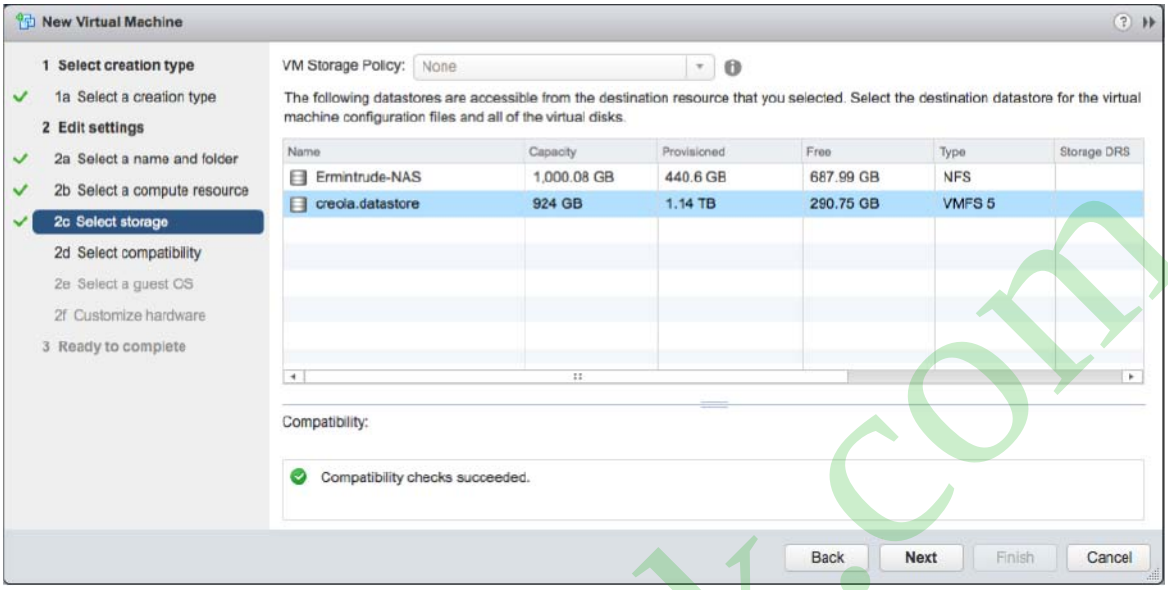


Step 8 Enter a **name** for the new Virtual Machine.

Step 9 Select a **location** for the new Virtual Machine, then click **Next**.

The dialog refreshes, showing options for selecting the datastore.(Figure 2-5 on page 11).

Figure 2-5: New Virtual Machine dialog (selecting the datastore)

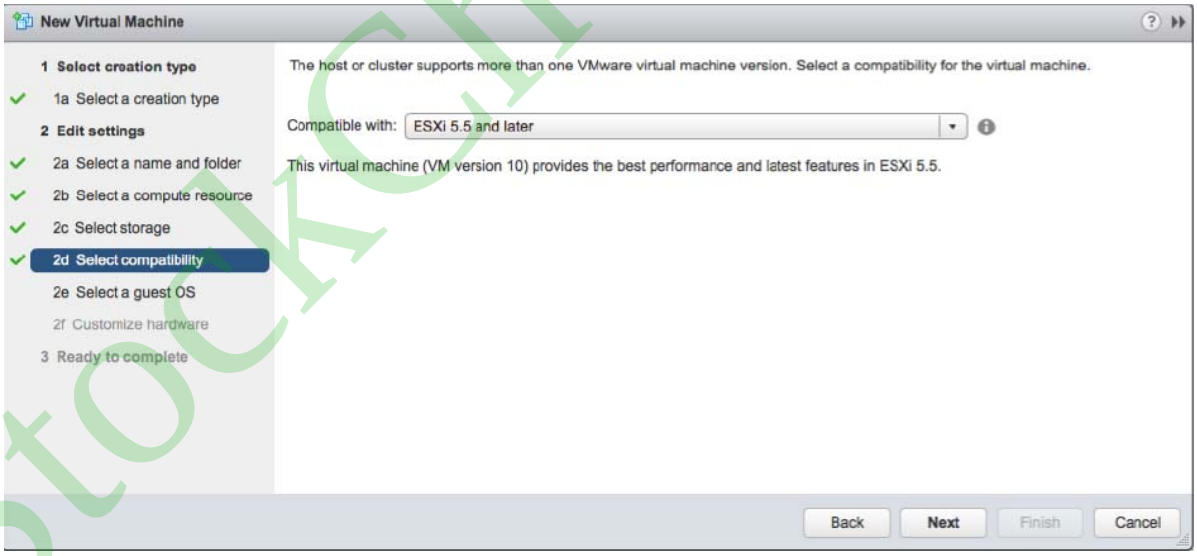


Step 10 Select the **datastore** for the new Virtual Machine configuration files and all of the virtual disks.

Step 11 Click **Next**.

The dialog refreshes, showing compatibility options (Figure 2-6).

Figure 2-6: New Virtual Machine dialog (compatibility options)



Step 12 Using the Compatible with menu, select the **ESXi compatibility** for the new Virtual Machine.

Note When adding the VMDK to ESX6, it treats this as sparse by default, whereas in ESX 5 it is thick. Converting the vEOS VMDK file from thin to thick would allow it to boot properly in ESX6: vmkfstools -i vEOS-lab-4.18.5M.vmdk -d eagerzeroedthick vEOS-lab-4.18.5M-thick.vmdk

Note If the VM keeps rebooting and showing “This is not a bootable disk. Please insert a bootable floppy and press any key to try again" then refer this link:
https://eos.arista.com/common-issues-when-deploying-cvx-4-18-2fon-vcenter-6-or-6-5/

Please refer following links for details about the issue and solution.

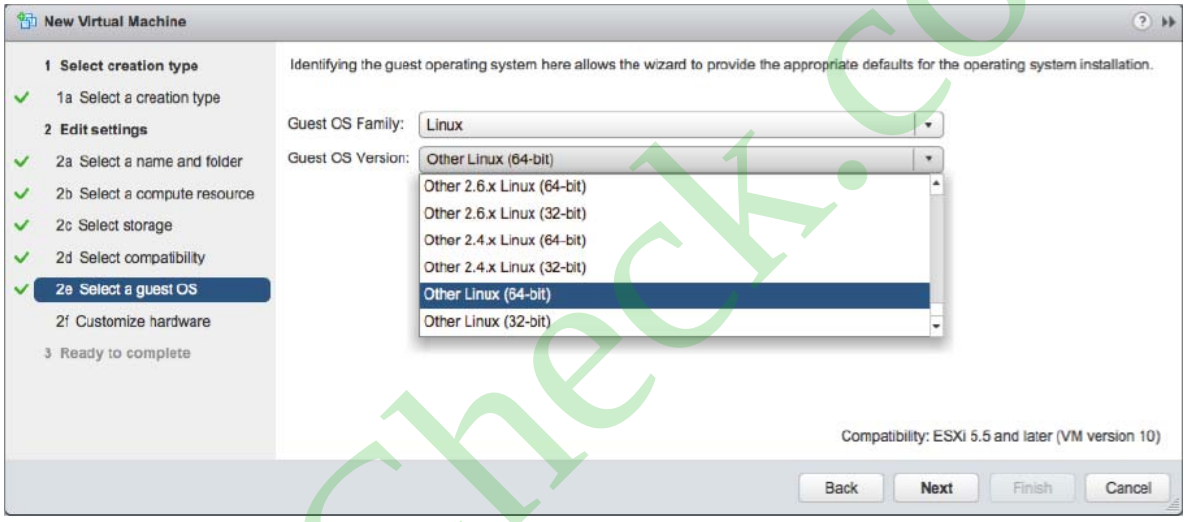
https://eos.arista.com/tip-for-arista-veos-on-vmware-esx-6/

https://eos.arista.com/common-issues-when-deploying-cvx-4-18-2f-on-vcenter-6-or-6-5/

Step 13 Click **Next**.

The dialog refreshes, showing operating system selection options (Figure 2-7 on page 12).

Figure 2-7: New Virtual Machine dialog (operating system options)



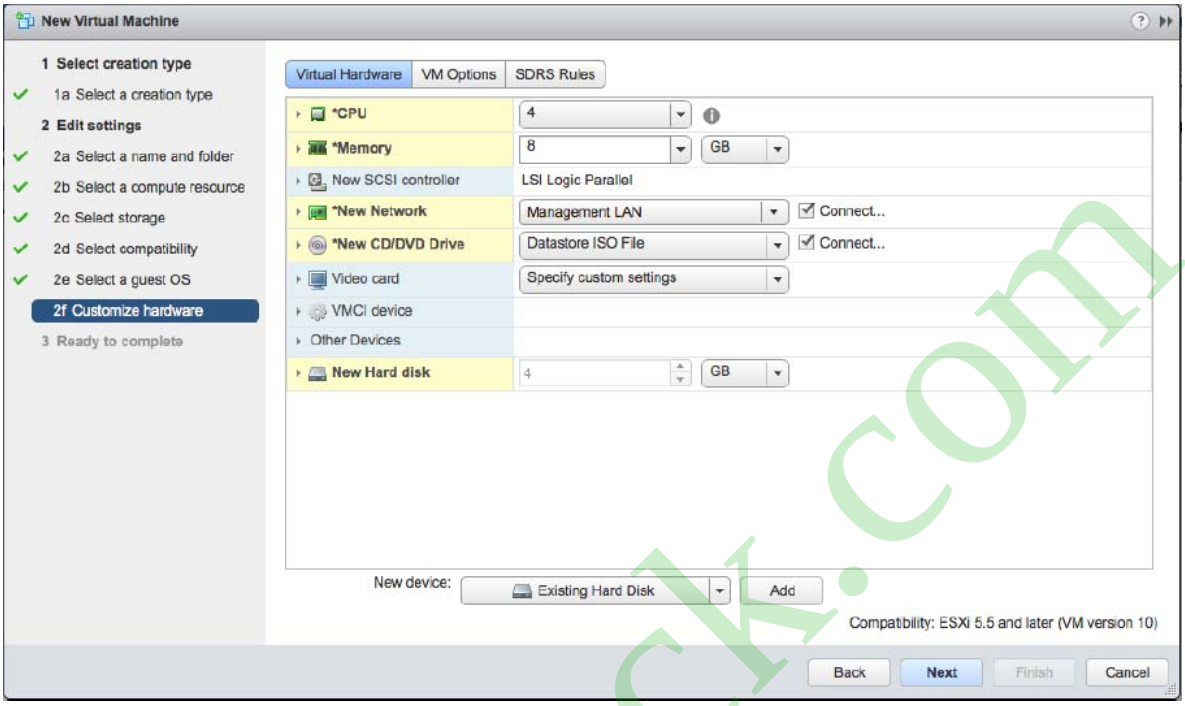
Step 14 Using the Guest OS Family menu, choose **Linux**.

Step 15 Using the Guest OS Version menu, choose **Other Linux (64--bit)**.

Step 16 Click **Next**.

The dialog refreshes, showing options for customizing hardware (Figure 2-8).

Figure 2-8: New Virtual Machine dialog (hardware configuration options)



Step 17 Change the default settings for the following options:

CPU	Set to **4** (number of CPUs)
Memory	Set to **8 GB**
New Hard Disk	Delete the current setting (leave this option empty).
New Network	Specify connection to Network LAN segment with connectivity to CVX client devices (the Management LAN). This connection is used for CVX client / server communications.
New CD/DVD Drive	Select **Datastore ISO File**, and specify the **Aboot-veos-8.0.0.iso** you downloaded in step 3. Make sure the **Connect on boot** option is selected.
Existing Hard Disk	Specify the **EOS--4.16.8M.vmdk** you downloaded in step 4.

Step 18 (Optional) Delete the floppy drive and SCSI controller.

Step 19 Click **Next**.

You are now ready to begin the CVX configuration (see "CVX Configuration").

2.4 CVX Configuration

CVX, its clients, and its services, are independently configured. These sections describe configuration processes for each:

- "Ports Used by CVX"
- "CVX Server Configuration"
- "CVX Client Configuration" on page 15
- "CVX Client Services Configuration" on page 17

2.4.1 Ports Used by CVX

CVX uses the following ports:

- Controller database (Controllerdb): Port 9979
- Client-server out-of-band connection: Port 50003
- CVX cluster peer out-of-band connection: Port 50004

Note All of these connections are TCP.

2.4.2 CVX Server Configuration

Enabling CVX on the CVX Server

CVX parameters for the server infrastructure are configured in CVX configuration mode. CVX configuration mode is not a group-change mode; *running-config* is changed when commands are entered, and exiting the mode does not modify *running-config*. The **cvx** command places the switch in CVX configuration mode.

CVX is disabled by default. The **no shutdown (CVX)** command enables CVX on the switch.

Example

- These commands enter CVX-configuration mode and enable CVX.

```
switch(config)#cvx
switch(config-cvx)#no shutdown
switch(config-cvx)#
```

CVX Heartbeat Configuration

CVX synchronizes with its client devices by exchanging heartbeat signals. The heartbeat transmission frequency and timeout period determine when a client's access to the server is disrupted.

The interval between heartbeat messages that the server transmits is specified by the **heartbeat-interval (CVX)** command. The CVX timeout period is specified by the **heartbeat-timeout (CVX)** command. When CVX does not receive a subsequent heartbeat message from a CVX client before the timeout expiry, the server discontinues CVX services to that client.

Best practices dictate that CVX and its client applications configure identical heartbeat interval and heartbeat timeout values.

Example

- These commands configure a CVX heartbeat interval of 30 seconds and a server heartbeat timeout period of 90 seconds.

```
switch(config-cvx)#heartbeat-interval 30
switch(config-cvx)#heartbeat-timeout 90
switch(config-cvx)#
```

Disabling CVX on the CVX Server

Important! Before disabling or de-configuring CVX on the CVX server, CVX client services should be explicitly disabled or shut down. Failure to disable or de-configure services prior to disabling or de-configuring CVS may result in CVX features continuing to run after CVX shutdown.

When disabling the CVX service, service VXLAN configuration may be retained or erased. Be sure to disable or shut down client services prior to disabling the CVX service.

Examples

- These commands shut down the CVX service while retaining the CLI configuration for service VXLAN.

```
localhost(config)#cvx
localhost(config-cvx)#service vxlan
localhost(config-cvx-vxlan)#shutdown
```

- These commands shut down the CVX service and also erase service VXLAN CLI configuration.

```
localhost(config-cvx-vxlan)#
localhost(config)#cvx
localhost(config-cvx)#no service vxlan
```

2.4.3 CVX Client Configuration

This section describes the CVX client configuration and commands that enable CVX services. Most commands for the configuration of the CVX client infrastructure are accessed in Management-CVX configuration mode.

2.4.3.1 Enabling CVX on the CVX Client

CVX client parameters are configured in Management-CVX configuration mode. Management-CVX configuration mode is not a group-change mode; *running-config* is changed when commands are entered, and exiting the mode does not modify *running-config.* The **management cvx** command places the switch in Management-CVX configuration mode.

CVX client is disabled by default. The **no shutdown (Management-CVX)** command enables CVX client on the switch.

For the CVX network topology service to create an inventory of all CVX clients, ensure that LLDP is enabled on each client switch using the **lldp run** command.

Example

- These commands enter Management-CVX-configuration mode and enable the CVX client.

```
switch(config)#lldp run
switch(config)#management cvx
switch(config-mgmt-cvx)#no shutdown
switch(config-mgmt-cvx)#
```

2.4.3.2 CVX Client Heartbeat Configuration

A CVX client synchronizes and maintains contact with CVX by exchanging heartbeat signals. The heartbeat transmission frequency and timeout period define when communication with CVX will be considered down.

The interval between heartbeat messages that the CVX client transmits is configured by the **heartbeat-interval (Management-CVX)** command.

The CVX client timeout period is specified by the **heartbeat-timeout (Management-CVX).** When a CVX client does not receive a subsequent heartbeat message from CVX within this timeout period, the client assumes that services provided by CVX are no longer available.

Best practices dictate that a CVX client's heartbeat interval and heartbeat timeout values are identical to those of the CVX server to which it connects.

Example

- This command configures a CVX client heartbeat interval of 30 seconds and client timeout period of 90 seconds.

```
switch(config-mgmt-cvx)#heartbeat-interval 30
switch(config-mgmt-cvx)#heartbeat-timeout 90
switch(config-mgmt-cvx)#
```

2.4.3.3 Connecting the CVX Client to a Server

The **server host (Management-CVX)** command identifies the location of the CVX server that the client accesses. The **source-interface (Management-CVX)** command specifies the interface from which the client derives the IP address it uses as the source in CVX packets that it transmits. And the **no shutdown (Management-CVX)** command enables CVX on the client switch.

Example

- These commands configure the switch as a CVX client, connecting to a CVX server at IP address 10.1.1.14 and using IP address 10.24.24.1 as the source address for its outbound packets.

```
switch(config)#interface loopback 5
switch(config-if-Lo5)#ip address 10.24.24.1/24
switch(config-if-Lo5)#management cvx
switch(config-mgmt-cvx)#server host 10.1.1.14
switch(config-mgmt-cvx)#source-interface loopback 5
switch(config-mgmt-cvx)#no shutdown
switch(config-mgmt-cvx)#
```

2.4.4 CVX Client Services Configuration

Switches running EOS must be configured as CVX clients to access the network services running on CVX. Individual services may require additional configuration.

Refer to the following for information regarding the services available to a CVX client.

- “Configuring OpenStack Service” on page 17
- “Configuring VXLAN Control Service” on page 17
- “Configuring Hardware Switch Controller Service (HSC)” on page 18
- “Configuring Network Topology Service” on page 19

2.4.4.1 Configuring OpenStack Service

The OpenStack service is enabled from CVX-OpenStack configuration mode, which is accessed by the **service openstack** command. The **no shutdown (CVX-OpenStack)** command enables CVX OpenStack services on the CVX server. Additional configuration is necessary to deploy OpenStack; http://docs.openstack.org/.

Example

- These commands enable the CVX-OpenStack service.

```
switch(config-cvx)#service openstack
switch(config-cvx-openstack)#no shutdown
switch(config-cvx-openstack)#
```

2.4.4.2 Configuring VXLAN Control Service

The VXLAN control service is enabled on CVX by the **no shutdown (CVX-VXLAN)** command and on the client switches by enabling CVX and configuring the VXLAN as a controller client. When VXLAN control service is enabled, CVX functions as a VXLAN controller for its clients.

For information about configuring VXLAN on the client switch, see the VXLAN chapter of the *User Manual*.

Examples

- These commands enable VXLAN control service on the CVX server.

```
switch(config-cvx)#service vxlan
switch(config-cvx-vxlan)#no shutdown
switch(config-cvx-vxlan)#
```

- These commands enable VXLAN Control Service on the CVX client. (This example assumes that the VXLAN has already been configured on the client switch. For information about configuring VXLAN, see the VXLAN chapter of the *User Manual*).

```
switch(config)#interface vxlan 1
switch(config-if-Vx1)#vxlan controller-client
```

2.4.4.3 Configuring Hardware Switch Controller Service (HSC)

The hardware switch controller (HSC) service is enabled on the CVX server by the **no shutdown (CVX-HSC)** command.

Certificate Requirements for CVX Interoperability with VMware NSX 6.2.2 and higher

The certificate type needs to be changed from MD5 to SHA512 for use with VMware NSX 6.2.2. Complete the following steps to make the change.

Step 1 At the EOS prompt of CVX, use the following commands.

```
switch(config)#cvx
switch(config-cvx)#service hsc
switch(config-cvx-hsc)#shut
```

Step 2 Acquire superuser privileges and edit the default.

```
switch(config)#bash
switch(config)#sudo su
switch(config)#vi /usr/bin/ovs-pki
```

Step 3 Find and replace **default_md** with **sha512 (**from **md5)**

```
default_md =md5
default_md =sha512
```

Step 4 Delete all files and folders from **/persist/secure/openvswitch/**

```
cd /persist/secure/openvswitch/
bash-4.1#sudo rm -r *
```

Step 5 Generate the new certificate

```
[admin@CVX ~]$ exit
logout
CVX(config-cvx-hsc)#no sh
CVX(config-cvx-hsc)#end
```

Step 6 Verify the change using the command:

```
CVX# show nsx status
```

Example

- These commands enable the CVX-HSC service.

```
switch(config)#cvx
switch(config-cvx)#no shutdown
switch(config-cvx)#service hsc
switch(config-cvx-hsc)#no shutdown
```

The HSC service sends flood lists to each VTEP through CVX. Some controllers (such as VMware NSX's Service Nodes) implement replication nodes for head-end replication of unknown packets. For these controllers, BUM packets should be sent to a single replication node (send-to-any replication), and the flood list sent by the HSC service is a list of replication nodes. Other controllers (such as Nuage VSP) require each VTEP to perform its own head-end replication. For these, BUM packets should be sent to every known VTEP, and the flood list sent by the HSC service is the list of VTEPs.

The default behavior is to use a send-to-any replication list of VTEPs. If the required behavior is send-to-all replication of, use the **all** option of the **vtep (CVX-HSC)** command.

Example

- This command configures the CVX-HSC service to use send-to-any replication.

```
switch(config-cvx-hsc)#vtep flood list type all
switch(config-cvx-hsc)#
```

Important! HSC also makes use of the VXLAN control service; ensure that VXLAN control service is enabled and properly configured (see **VXLAN Control Service** for details).

HSC also requires a connection to an OVSDB controller. Configure the IP address or host name of the controller using the **manager** command.

Example

- This command configures the CVX-HSC service to connect to an OVSDB controller at IP address 192.168.2.5, using the default port 6632.

```
switch(config-cvx-hsc)#manager 192.163.2.5
switch(config-cvx-hsc)#
```

Having established a connection to the OVSDB controller, the HSC service will publish the inventory of switches managed by CVX to OVSDB. For the inventory to succeed, LLDP must be enabled on each CVX client switch with the **lldp run** command.

Note LLDP is enabled by default on Arista switches

Example

- This command enables LLDP.

```
switch(config)#lldp run
switch(config)#
```

2.4.4.4 Configuring Network Topology Service

A network topology agent runs on each Arista switch whether or not the switch is connected to a CVX server. It requires no configuration. The network topology service on the CVX server is also enabled by default and requires no configuration.

To view the aggregated topology information, use the **show network physical-topology** command on the switch running the CVX server instance.

Examples

- This command displays all visible hosts.

```
switch#show network physical-topology hosts

Unique Id            Hostname
-------------------- ------------------------------
001c.7385.be69       cvx287.sjc.aristanetworks.com
0000.6401.0000       cvc1
0000.6402.0000       cvc2
0000.6403.0000       cvc3
0000.6404.0000       cvc4
bcf6.85bd.8050       dsj14-rack14-tor1
```

- This command displays all connections in the topology.

```
switch#show network physical-topology neighbors

cvx287.sjc.aristanetworks.com
Interface Neighbor Intf Neighbor Host
----------------- ----------------- ----------------------
Ethernet1         Ethernet7 cvc4
Ethernet2         Ethernet7 cvc2
Ethernet9         Ethernet7 cvc1
Ethernet10        Ethernet7 cvc3
Management1 27    dsj14-rack14-tor1

                <-------OUTPUT OMITTED FROM EXAMPLE-------->

dsj14-rack14-tor1
Interface Neighbor Intf Neighbor Host
----------------- ----------------- ----------------------
27 Management1    cvx287.sjc.aristanetwork
```

Related Topics

- “CVX Secure out-of-band Connection” on page 20
- “CVX High Availability” on page 24
- “CVX Command Descriptions” on page 34
- “CVX Overview” on page 4
- “CVX Services” on page 5
- “Deploying CVX” on page 6

2.5 CVX Secure out-of-band Connection

This feature adds support for securing out-of-band connection between CVX server and CVX clients by SSL/TLS transport protocol. SSL/TLS is an application-layer protocol that provides secure transport between client and server through a combination of authentication, encryption and data integrity. SSL/TLS uses certificates and private-public key pairs to provide this security. We will use the term SSL to mean SSL/TLS.

By default, CVX server and CVX clients communicate over insecure transport (there is no authentication and encryption between CVX server and CVX clients). This poses the possibility of security risks, such as communicating with untrusted CVX server and CVX clients, or eavesdropping CVX server/client communications. This feature can be used to secure the out-of-band connection between CVX server and CVX clients.

Note The CVX client-server out-of-band connection uses port 50003. The CVX cluster peer out-of-band connection uses port 50004. These are TCP ports.

For more information, see:

- “Configuring the CVX Secure out-of-band Connection” on page 21
- “Show Commands” on page 22
- “Troubleshooting” on page 24

2.5.1 Configuring the CVX Secure out-of-band Connection

This feature uses SSL certificate and key management infrastructure for managing certificates, keys and SSL profiles. For more information regarding this infrastructure see SSL Certificate and Key Management in the Arista User's Guide.

Step 1 On CVX server, copy the server certificate and key and also the CA certificate to verify CVX clients.

```
switch(config)#!Copy the PEM encoded certificate and RSA key files for CVX
server
switch(config)#!Lets call them server.crt and server.key
switch(config)#copy <url> certificate:server.crt
switch(config)#copy <url> sslkey:server.key
switch(config)#!Copy the PEM encoded CA certificate to verify the certificate
of CVX clients.Lets call it ca.crt
switch(config)#copy <url> certificate:ca.crt
```

Step 2 On CVX server, configure SSL profile with the certificates and key as below. Lets call the SSL profile as "serverssl".

```
switch(config)#management security
switch(config-mgmt-security)#ssl profile serverssl
switch(config-mgmt-sec-ssl-profile-serverssl)#certificate server.crt key
server.key
switch(config-mgmt-sec-ssl-profile-serverssl)#!You can trust multiple CA
certificates
switch(config-mgmt-sec-ssl-profile-serverssl)#trust certificate ca.crt
```

Note If you are using intermediate certificates to build a 'Chain of Trust' (such as server.crt -> intermediate1.crt -> intermediate2.crt -> ca.crt), then you need to configure the intermediate certificates as part of the SSL profile using the following commands:

```
switch(config-mgmt-sec-ssl-profile-serverssl)#chain certificate intermediate1.crt
switch(config-mgmt-sec-ssl-profile-serverssl)#chain certificate intermediate2.crt
```

Step 3 On CVX server, configure to use the "serverssl" SSL profile. With this configuration, the CVX server starts listening on a secure port. The CVX server will continue to listen on the default port. i.e., the CVX server will accept connections from CVX clients over both SSL and default non-SSL transports. During a SSL negotiation, the CVX server will authenticate itself to the CVX clients by presenting 'server.crt' and it verifies the authenticity of the CVX client by checking if the CVX client certificate is signed by the trusted certificate "ca.crt".

```
switch(config)#cvx
switch(config-cvx)#ssl profile serverssl
```

Step 4 On CVX client, copy the client certificate and key and also the CA certificate to verify CVX server.

```
switch(config)#!Copy PEM encoded certificate and RSA key files for CVX client
switch(config)#!Lets call them client.crt and client.key
switch(config)#copy <url> certificate:client.crt
switch(config)#copy <url> sslkey:client.key
switch(config)#!Copy PEM encoded CA certificate used to verify the
switch(config)#!certificate of CVX server. Lets call it ca.crt
switch(config)#copy <url> certificate:ca.crt
```

Note If you are using intermediate certificates to build a 'Chain of Trust' (such as client.crt -> intermediate1.crt -> intermediate2.crt -> ca.crt), then you need to configure the intermediate certificates as part of the SSL profile using the following commands:

switch(config-mgmt-sec-ssl-profile-clientssl)#**chain certificate intermediate1.crt**
switch(config-mgmt-sec-ssl-profile-clientssl)#**chain certificate intermediate2.crt**

Step 5 On CVX client, configure SSL profile with the certificates and key as below. Lets call the SSL profile as “clientssl”.

```
switch(config)#management security
switch(config-mgmt-security)#ssl profile clientssl
switch(config-mgmt-sec-ssl-profile-clientssl)#certificate client.crt key
client.key
switch(config-mgmt-sec-ssl-profile-clientssl)#!You can trust multiple CA
certificates
switch(config-mgmt-sec-ssl-profile-clientssl)#trust certificate ca.crt
```

Step 6 On CVX client, configure to use the SSL profile – “clientssl”. With this configuration, the CVX client will connect to the secure port of the CVX server over SSL transport. During SSL negotiation, the CVX client will authenticate itself to the CVX server by presenting ‘client.crt’ and it verifies the authenticity of the CVX server by checking if the CVX server certificate is signed by the trusted certificate ‘ca.crt’.

```
switch(config)#management cvx
switch(config-mgmt-cvx)#ssl profile clientssl
```

2.5.2 Show Commands

For information regarding show commands of SSL certificate, key and profile, please refer to SSL Certificate and Key Management.

To show the SSL profile status on CVX server, use the show cvx command.

```
switch#show cvx

CVX Server
 Status: Enabled
 UUID: beb19142-dfaa-11e4-b996-001c73105347
 Heartbeat interval: 20.0
 Heartbeat timeout: 60.0
 SSL profile: serverssl
   Status: Enabled
```

The “Enabled” SSL status means that the SSL profile is enabled for CVX server and the CVX clients can connect to CVX server over SSL transport. If there are any errors, then the status will show “Disabled” and the reason will be listed. In ‘Disabled’ state, the CVX clients wont be able to connect to CVX server over SSL transport.

To show the SSL connection status of CVX clients on CVX server, use the **show cvx connections** command.

```
switch#show cvx connections

Switch 00:1c:73:10:53:48
 Hostname: sq302
 Status: up
 Last heartbeat sent: 0:00:04 ago
 Last heartbeat received: 0:00:10 ago
 Clock offset: -0.00201620385865
 Out-of-band connection: SSL secured
 In-band connection: Not secured (SSL not supported)
```

The out-of-band connection shows as “SSL secured”, which means that the CVX client has connected to CVX server over SSL transport. The in-band connection is another connection between CVX server and CVX client. The SSL is not yet supported for this connection and hence it shows as ‘SSL not supported’. There is already some level of protection for the in-band connection. The CVX server and CVX client opens up the access to in-band connection only if the out-of-band connection is successful. Since the out-of-band connection is configured to use SSL, the in-band connection access is granted only for authentic CVX client and CVX server.

To show SSL profile status and connection status on CVX client, use ‘show management cvx’ command

```
switch#show management cvx

CVX Client
 Status: Enabled
 Last connected time: 2015-04-14 11:16:19
 Connection status: Connected
    Out-of-band connection: SSL secured
    In-band connection: Not secured (SSL not supported)
 Negotiated version: 2
 Controller UUID: 0e7dee2e-e2cf-11e4-880f-001c73105347
 Controller: 127.0.0.1
    Last heartbeat sent: 0:00:00 ago
    Last heartbeat received: never
    Clock offset: 0.0
 SSL profile: clientssl
    Status: Enabled
```

The “Enabled” SSL status means that the SSL profile is enabled and the CVX client can connect to CVX server over SSL transport. If there are any errors, then the status will show as “Disabled” and the reason will be listed. In Disabled state, the CVX client won’t be able to connect to the CVX server.

Similar to the CVX server, the out-of-band connection shows as “SSL secured” and the SSL is not yet supported for in-band connection.

The possible reasons for ‘Disabled’ SSL status on CVX server and CVX client are:

- **SSL profile does not exist:** If the SSL profile configured under CVX server/client is not configured under ‘management security’, you will see this message. Please configure the SSL profile with required certificates and key under ‘management security’.
- **Invalid SSL profile:** If the SSL profile configured under CVX server/client is in ‘invalid’ state, you will see this message. Check ‘show management security ssl profile <name>’ command to see the errors on the SSL profile and fix them.
- **Trusted certificates not configured in SSL profile:** If the SSL profile configured under CVX server/client does not have trusted certificates configured, you will see this message. Please configure trusted CA certificates in the SSL profile.

- **Certificate not configured in SSL profile:** If the SSL profile configured under CVX server/client does not have certificate key pair configured, you will see this message. Please configure certificate and key pair in the SSL profile.
- **Diffie-Hellman parameters not yet ready:** When EOS is booted, a Diffie-Hellman parameters file is auto generated by the system if one does not exist. This Diffie-Hellman parameters file is used for symmetric key exchange during SSL negotiation. Only the CVX server uses this file and hence this message can be seen only on 'show cvx' command output. If the file is not yet generated, you will see this message. When the file is ready, this message automatically goes away and the SSL profile will become 'Enabled'.

2.5.3 Troubleshooting

Check show cvx on the CVX server and see if the SSL profile is in "Enabled" state. If it's in "Disabled" state, check the reason listed and fix it.

Check "show management cvx" on CVX client and see if SSL profile is in "Enabled" state. If it's in "Disabled" state, check the reason listed and fix it.

Related Topics

- "CVX High Availability" on page 24
- "CVX Command Descriptions" on page 34
- "CVX Overview" on page 4
- "CVX Services" on page 5
- "Deploying CVX" on page 6
- "CVX Configuration" on page 14

2.6 CVX High Availability

CVX provides high availability by enabling you to use multiple (redundant) CVX Controllers in the same cluster. Each Controller in the cluster has its own dedicated machine so that if a Controller fails, the failure is isolated to a single machine.

Within a cluster, one of the Controllers is a primary (leader), and the other Controllers are backup (follower) Controllers. If the primary Controller fails, one of the backup Controllers automatically assumes the role of the primary Controller.

CVX high availability does not prevent or compromise the detection of software failures or link failures that may cause Controllers to be unreachable on the network.

The configuration that is required to ensure CVX is set up for high availability involves:

- Configuring the CVX cluster.
- Configuring the CVX clients.

For more information, see:

- "CVX Clusters" on page 25
- "Handling of CVX Controller Failures" on page 26
- "CVX Support for EOS Failure Modes" on page 26
- "Client Interaction" on page 27
- "Service Agents Interaction" on page 27
- "Leader Election" on page 27

2.6.1 CVX Clusters

CVX clusters are sets of CVX Controllers (usually 3 Controllers). Within a cluster, each Controller runs on its own dedicated machine, and all of the Controllers run the same version of CVX. Each Controller in the cluster functions as either the primary (leader) Controller, or a backup (follower) Controller.

One of the CVX Controllers is elected by the group of Controllers to be the primary Controller. Once a Controller is elected to be the primary, the other Controllers in the cluster are automatically assigned the role of backup Controllers. Cluster members maintain an out-of-band connection amongst themselves, which is used for the leader election protocol.

Note CVX Controllers in a cluster that are not the primary Controller always function as backup Controllers. Within the same cluster, only one CVX Controller can assume the role of a primary at any time.

For more information, see:

- “Required Number of Controllers to Support High Availability” on page 25
- “Cluster Configuration Options” on page 25

2.6.1.1 Required Number of Controllers to Support High Availability

A cluster must have enough Controllers so that in the case of a failure of the primary Controller, there are enough remaining Controllers for the election process to be completed. The election process is used by clusters to select a new primary Controller in the case of failure.

Note The number of Controllers for a cluster is **3** (one primary and two backup Controllers).

Examples

In a cluster with only **two** Controllers (one primary and one backup), a simple majority of backup Controllers does not exist after a failure of the primary Controller. A simple majority of two backup Controllers is required for the leader election process.

Related Topics

- “Cluster Configuration Options” on page 25
- “Handling of CVX Controller Failures” on page 26
- “CVX Support for EOS Failure Modes” on page 26
- “Client Interaction” on page 27
- “Service Agents Interaction” on page 27
- “Leader Election” on page 27

2.6.1.2 Cluster Configuration Options

You can configure the cluster for high availability using either of the following modes:

- Cold followers mode - Only the Controllerdb of the primary (leader) CVX Controller mounts from the client switches.
- Warm followers mode - The Controllerdb of every (all) CVX Controllers in the cluster mount from the client switches.

Advantages and disadvantages of the modes

The advantage of the warm follower mode is that if the primary CVX Controller fails, the switchover to the new primary is faster than a switchover in cold follower mode. The reason for this is that the state of the new primary does not have to be rebuilt from scratch. The disadvantage of the warm follower mode is that serialization from the switch is slower compared to cold follower mode.

Related Topics

- "Required Number of Controllers to Support High Availability" on page 25
- "Handling of CVX Controller Failures" on page 26
- "CVX Support for EOS Failure Modes" on page 26
- "Client Interaction" on page 27
- "Service Agents Interaction" on page 27
- "Leader Election" on page 27

2.6.2 Handling of CVX Controller Failures

CVX Controllers can fail because of hardware or software faults. Because EOS agents are designed to be software fault-tolerant, an agent that fails is automatically restarted and resumes operation statefully. The most recent saved state in Sysdb for the agent is used to restore the state of the agent.

Unlike software failures, hardware failures are not handled by EOS. CVX handles hardware failures through the use of redundant backup (follower) CVX Controllers that run on their own dedicated machine. Within a cluster, any backup Controller can assume the role of the primary (leader) Controller.

Note In the event of a network partition, the partition with a majority of the Controllers elects a leader from its Controllers, and the minority partition relinquishes any leadership it might have had.

Related Topics

- "CVX Support for EOS Failure Modes" on page 26
- "Client Interaction" on page 27
- "Service Agents Interaction" on page 27
- "Leader Election" on page 27
- "CVX Clusters" on page 25

2.6.3 CVX Support for EOS Failure Modes

CVX supports both EOS failure modes that apply when a CVX Controller fails. The EOS failure modes are:

- Fail-stop
- Fail-recover

Because CVX supports both EOS failure modes, a failed CVX Controller can rejoin the cluster if the following failures occur:

- A crash of the agent or machine running CVX.
- The CVX controller or dedicated machine it runs on is removed (partitioned) from the cluster.

Related Topics

- "Client Interaction" on page 27

- "Service Agents Interaction" on page 27
- "Leader Election" on page 27
- "CVX Clusters" on page 25
- "Handling of CVX Controller Failures" on page 26

2.6.4 Client Interaction

Client switches maintain an out-of-band connection to all members of the cluster. The connection is used to determine liveness and for communications. The connection is also used to signal a change in leadership (switchover) to the client switches. Switchovers that are changes in leadership within a cluster are executed similarly to CVX Graceful Reboot switchovers.

The ControllerClient agent on the switch is responsible for maintaining liveness with the Controllers and for exchanging metadata. The ControllerClient agent registers with all cluster members. Each Controller's ControllerStatus has an additional flag to record whether the Controller is a leader within the cluster.

If there is more than one leader, the switch automatically waits until only one Controller is designated as the leader in the cluster. Once a single Controller is designated as the leader, the switch executes a graceful switchover to the new leader Controller.

Related Topics

- "Service Agents Interaction" on page 27
- "Leader Election" on page 27
- "CVX Clusters" on page 25
- "Handling of CVX Controller Failures" on page 26
- "CVX Support for EOS Failure Modes" on page 26

2.6.5 Service Agents Interaction

One change to Service Agents is required to support CVX high availability. Service Agents must be modified to include the leader flag (this flag identifies the leader CVX Controller in the cluster). On a leader switchover, Service Agents are deactivated on the old leader Controller and activated on the new leader Controller. The client switches will perform a graceful switchover to the new leader Controller.

Related Topics

- "Leader Election" on page 27
- "CVX Clusters" on page 25
- "Handling of CVX Controller Failures" on page 26
- "CVX Support for EOS Failure Modes" on page 26
- "Client Interaction" on page 27

2.6.6 Leader Election

Leader election is an internal, system-run process that is essential to CVX high availability. The leader election process is used to safely elect a new leader Controller within a cluster following the failure of the current leader Controller, or a network configuration change that results in the loss of the current leader Controller in the cluster.

The leader election process is designed to ensure stability of leader Controllers within clusters. The process is based on an algorithm that provides the mechanism for the backup (follower) Controllers to elect (by consensus), the new leader Controller in the cluster.

2.6.7 Configuring CVX Clusters for High Availability

Configuring CVX clusters for high availability is a simple process that involves pointing each cluster member to the other cluster members using the **`peer host`** command. The objective of this task is to successfully register each cluster member with the other cluster members. Successful registration of the cluster members with each other ensures that the members can communicate with each other to elect a new leader member if the original leader member fails.

Once you complete the process, the cluster members will be successfully registered with each other. In addition, the cluster members will automatically elect a leader member and assign the 'leader' to that member. The non-leader members are automatically assigned the role of 'follower'.

Requirements

The requirements for setting up clusters for high availability are:

- The number of CVX Controllers in a cluster is **3**.
- An **odd number** of CVX instances (CVX Controllers) are required to form a cluster.

Note If an even number of CVX Controllers are configured in a cluster, a CVX instance will automatically refuse to participate in the cluster.

- All cluster members must point to each other. This is essential for clusters to operate normally. (The steps required to complete this task are included in the following procedure.)

Procedure

Note This procedure provides configuration examples for each step. The 'example' cluster used throughout the procedure contains 3 cluster members (named *cvs1*, *cvs2*, and *cvs3*). The IP addresses of the cluster members are:
- *cvs1* (10.0.0.1)
- *cvs2* (10.0.0.2)
- *cvs3* (10.0.0.3).

Complete the following steps to configure clusters for high availability.

Step 1 Using the **`peer host`** command, configure one of the cluster members to point to every other cluster member.

This example shows the configuration of cluster member ***cvs1*** to point to the other cluster members (*cvs2* and *cvs3*).

```
cvs1(config-cvx)#peer host 10.0.0.2 (connects cvs1 to cvs2)
cvs1(config-cvx)#peer host 10.0.0.3 (connects cvs1 to cvs3)
```

Step 2 Use the **`sh cvx`** command to check the **Mode** and **Peer registration state** status values for cluster member *cvs1*. The status values should be:

- **Mode** = *Cluster*
- **Peer registration state** = *Connecting*

Note **Mode** automatically changes from “Standalone” to “Cluster” when configuring a CVX cluster. This is because the presence of multiple CVX “peers” causes the Mode to change to “Cluster”.

Peer registration state remains in “Connecting” status after you configure the first cluster member. This is because the two peers must register with each other for the registration of the two members to be successful.

Step 3 Using the **peer host** command, configure peer cluster member *cvs2* to point to every other cluster member.

This example shows the configuration of cluster member ***cvs2*** to point to the other cluster members (*cvs1* and *cvs3*).

```
cvs2(config-cvx)#peer host 10.0.0.1 (connects cvs2 to cvs1)
cvs2(config-cvx)#peer host 10.0.0.3 (connects cvs2 to cvs3)
```

Step 4 Use the **sh cvx** command to check the **Peer registration state** settings for *cvs1*. This is done to verify that peers *cvs1* and *cvs2* are successfully registered with each other.

```
cvs1(config-cvx)#sh cvx
```

Example

This example shows the output of the **sh cvx** command for *cvs1*. The **Peer registration state** setting of “Registration Complete” for peer *cvs2* indicates a successful registration between *cvs1* and *cvs2*.

```
cvs1(config-cvx)#sh cvx
CVX Server
  Status: Enabled
  UUID: 6c208fba-7324-11e5-8fef-1d98cdd3b27a
  Mode: Cluster
  Heartbeat interval: 20.0
  Heartbeat timeout: 60.0
  Cluster Status
    Name: default
    Role: Standby
    Leader: 10.0.0.2
    Peer timeout: 10.0
    Last leader switchover timestamp: 0:00:03 ago
    Peer Status for 10.0.0.3
      Peer registration state: Connecting
       Peer service version compatibility : Version mismatch
    Peer Status for 10.0.0.2
      Peer Id : 02-01-63-02-00-00
      Peer registration state: Registration complete
      Peer service version compatibility : Version ok
```

Step 5 Using the **peer host** command, configure peer cluster member *cvs3* to point to every other cluster member.

This example shows the configuration of cluster member ***cvs3*** to point to the other cluster members (*cvs1* and *cvs2*).

```
cvs3(config-cvx)#peer host 10.0.0.1 (connects cvs3 to cvs1)
cvs3(config-cvx)#peer host 10.0.0.2 (connects cvs3 to cvs2)
```

Step 6 Use the **sh cvx** command to check the **Peer registration state** settings for *cvs1*. This is done to verify that peers *cvs1* and *cvs3* are successfully registered with each other.

```
cvs1(config-cvx)#sh cvx
```

Example

This example shows the output of the **sh cvx** command for *cvs1*. The **Peer registration state** setting of “Registration Complete” for peer *cvs3* indicates a successful registration between *cvs1* and *cvs3*.

```
cvs1(config-cvx)#sh cvx
CVX Server
  Status: Enabled
  UUID: 6c208fba-7324-11e5-8fef-1d98cdd3b27a
  Mode: Cluster
  Heartbeat interval: 20.0
  Heartbeat timeout: 60.0
  Cluster Status
    Name: default
    Role: Standby
    Leader: 10.0.0.2
    Peer timeout: 10.0
    Last leader switchover timestamp: 0:05:37 ago
    Peer Status for 10.0.0.3
      Peer Id : 02-01-63-03-00-00
      Peer registration state: Registration complete
      Peer service version compatibility : Version ok
    Peer Status for 10.0.0.2
      Peer Id : 02-01-63-02-00-00
      Peer registration state: Registration complete
      Peer service version compatibility : Version ok
```

Next Steps

You are now ready to configure the CVX clients for high availability (see “Configuring CVX Clients for High Availability”).

2.6.8 Configuring CVX Clients for High Availability

Configuring CVX clients for high availability is a simple process that involves pointing each CVX client to every CVX cluster member using the **server host** command. The objective of this task is to successfully establish connections between each CVX client and every CVX cluster member. The connections are essential to ensure that the CVX clients are aware of the current status of each cluster member.

Important! If a CVX client is not pointing to every cluster member, or if it is pointing to a CVX instance (Controller) that is not part of the cluster, the client may not be aware of leadership changes in the cluster, or may become confused about which cluster member is currently the leader. Either of these scenarios can result in unexpected errors.

Once you complete the process, the CVX clients will have established connections with each cluster member (the Connection status for each Controller should be ‘Established’). In addition, the clients will be aware of which CVX instance (Controller) is currently the leader in the cluster.

Procedure

Note This procedure provides configuration examples for each step. The ‘example’ CVX client used throughout the procedure is named cvc1. The IP addresses of the cluster members are 10.0.0.1 (*cvs1*), 10.0.0.2 (*cvs2*), and 10.0.0.3 (*cvs3*).

Complete the following steps to configure CVX clients for high availability.

Step 1 Using the **server host** command, configure each of the CVX clients to point to every cluster member.

This example shows the configuration of client ***cvc1*** to point to all of the cluster members (the addresses of the cluster members are 10.0.0.1, 10.0.0.2, and 10.0.0.3).

```
cvc1(config-mgmt-cvx)#server host 10.0.0.1 (connects cvc1 to cluster member 10.0.0.1)
cvc1(config-mgmt-cvx)#server host 10.0.0.2 (connects cvc1 to cluster member 10.0.0.2)
cvc1(config-mgmt-cvx)#server host 10.0.0.3 (connects cvc1 to cluster member 10.0.0.3)
```

Step 2 Use the **sh man cvx** command to check the status of client ***cvc1***.

The Connection status for each cluster member should be “Established”. In addition, the client is also aware that cluster member 10.0.0.3 is the current Master.

```
cvc1(config-mgmt-cvx)#sh man cvx
CVX Client
  Status: Enabled
  Source interface: Inactive (Not configured)
  Controller cluster name: default
    Controller status for 10.0.0.1
      Connection status: established
        Out-of-band connection: Not secured
        In-band connection: Not secured (SSL not supported)
      Negotiated version: 2
      Controller UUID: 6c208fba-7324-11e5-8fef-1d98cdd3b27a
      Last heartbeat sent: 0:00:07 ago
      Last heartbeat received: 0:00:07 ago
    Controller status for 10.0.0.3
      Master since 0:03:34 ago
      Connection status: established
        Out-of-band connection: Not secured
        In-band connection: Not secured (SSL not supported)
      Negotiated version: 2
      Controller UUID: c64954b8-7324-11e5-9f33-51f8b016cae8
      Last heartbeat sent: 0:00:14 ago
      Last heartbeat received: 0:00:14 ago
    Controller status for 10.0.0.2
      Connection status: established
        Out-of-band connection: Not secured
        In-band connection: Not secured (SSL not supported)
      Negotiated version: 2
      Controller UUID: 6a0dbf2c-7324-11e5-94f3-ff17a8a1cdc8
      Last heartbeat sent: 0:00:05 ago
      Last heartbeat received: 0:00:05 ago
```

Related Topics

- “CVX Clusters” on page 25
- “Handling of CVX Controller Failures” on page 26
- “CVX Support for EOS Failure Modes” on page 26
- “Client Interaction” on page 27
- “Service Agents Interaction” on page 27
- “Leader Election” on page 27
- “Configuring CVX Clusters for High Availability” on page 28

2.7 CVX VIP

CVX VIP provides the virtual IP address that actively follows the master controller of the CVX cluster.

The virtual IP address of the CVX HA Cluster is configured on a macvlan interface setup on top of a physical management interface of the master controller. The virtual IP and virtual MAC needs to be provided by the customer as part of the controller configuration. This information is available to all controllers as each cluster member has to be configured manually by the user on all controllers.

The macvlan interface created should be designated as `Management0`. `Management0` is currently used for the ManagementActive interface on modular switches. Without explicit configuration of VIP and VMAC, CVX VIP functionality will not work in the CVX HA cluster.

Note Customers can pick the VMAC from a pool of MAC addresses reserved for use with CVX clusters. The OUI pool, 00:1C:73:00:00:AA – 00:1C:73:00:00:FF has been reserved for this purpose.

The macvlan interface is setup if all of the following conditions are met:

- VMAC is configured by the user
- The controller instance is a leader
- There are more than one controller instances
- The controller is not being run on a modular system

2.7.1 Configuring VIP

All CLI commands applicable to the management interface of the controller will be allowed on `Management0`, with the exception of layer 1 / phy level commands. So auto-negotiation or flow control can't be configured on the `Management0` interface. Instead these commands can only be run on the physical management interfaces. This makes sense as the phy-level configuration really depends on what the interface is physically wire

To configure VMAC/VIP

```
CVX(config)#interface management 0
CVX(config-if-Ma0)# mac-address 00:1C:72:00:00:FF
CVX(config-if-Ma0)# ip address 10.0.0.2
```

2.7.2 Data Replication

At EOS boot time, SSH host keys and Diffie-Hellman parameters are automatically generated and persistently stored on each controller. Multiple SSL profiles / keys / certificates might also be created and used by various agents on the controllers. Since these information contribute to the identity of the master, they will need to follow the master controller for all time.

In case of a controller switchover, the newly elected master controller will need to use the same SSH host keys & SSL profiles / keys / certificates to retain its identity and prevent any kind of network security alarms from being tripped. For example, if an SSH client notices that the host key has changed, it will normally flag an error warning the user of a possible man-in-the-middle type attack. Hence, this data will be replicated from the master to slaves.

2.7.3 SSH Host Key Tagging

SSH host keys are tagged with the chassis MAC address to deal with key regeneration issues when a supervisor module is moved from one chassis to another. This behavior will cause regeneration issues if we replicate the SSH host keys across the cluster resulting in the key fingerprint seen by management tools to be different.

To mitigate this, in addition to the chassis MAC address, the host keys would now be tagged with VMAC of the CVX HA cluster. If CVX VIP and VMAC are configured, SshHostKeysAgent will not regenerate keys if tagged VMAC and configured VMAC are the same, even if there is a mismatch between the chassis MAC and tagged MAC.

2.8 Upgrading CVX

You can upgrade CVX from a previous version to the current version by performing a few simple tasks. You can use the following procedure to upgrade any previous version of CVX to the current version.

2.8.1 Requirements

Make sure you follow these requirements during the upgrade process.

- If you have CVP, CVX and client switches in your environment, make sure you upgrade each component in the following order:
  - Upgrade CVP first
  - Upgrade the CVX cluster.
  - Upgrade the client switches. The reason for this is to ensure backward compatibility.
- You must upgrade the CVX cluster before you upgrade the client switches.
- If the CVX cluster is a 3 node cluster, make sure that only one node of the cluster is down at any one time during the upgrade process. (The order in which you upgrade the nodes does not matter.)

Pre-requisites

Before you begin the upgrade, make sure that:

- You perform a backup to ensure that you can restore data if needed.
- You download the latest version of CVX from Arista's Software Download page (https://www.arista.com/en/support/software-download).

Complete the following steps to upgrade CVX.

Step 1 Login to the cluster to be upgraded. (You can login to any node.)

Step 2 Upgrade the node. You must deploy a new image to perform the upgrade.

Step 3 Wait for the node you are upgrading to rejoin the cluster. Once the node has rejoined, go to the next step. (The node automatically rejoins the cluster as a follower node.)

Step 4 Repeat steps **1 through 3** to upgrade the two remaining nodes one node at a time. It does not matter the order in which you upgrade the remaining nodes.

2.9 CVX Command Descriptions

CVX Server Commands

- cvx, on page 35
- heartbeat-interval (CVX), on page 36
- heartbeat-timeout (CVX), on page 38
- port (CVX), on page 46
- show cvx, on page 52
- shutdown (CVX), on page 54

CVX Client Commands

- management cvx, on page 41
- heartbeat-interval (Management-CVX), on page 37
- heartbeat-timeout (Management-CVX), on page 39
- server host (Management-CVX), on page 48
- source-interface (Management-CVX), on page 59
- shutdown (Management-CVX), on page 58

CVX OpenStack Commands

- name-resolution force (CVX-OpenStack), on page 43
- name-resolution interval (CVX-OpenStack), on page 44
- service openstack, on page 50
- shutdown (CVX-OpenStack), on page 56

CVX VXLAN Control Service Commands

- resync-period, on page 47
- service vxlan, on page 51
- shutdown (CVX-VXLAN), on page 57
- vtep (CVX-VXLAN), on page 61

CVX Hardware Switch Controller (HSC) Commands

- manager, on page 42
- ovsdb-shutdown, on page 45
- shutdown (CVX-HSC), on page 55
- vtep (CVX-HSC), on page 60

CVX Network Topology Service Commands

- lldp run, on page 40
- show network physical-topology, on page 53

cvx

CVX (CloudVision eXtension) aggregates and shares status across a network of physical switches running EOS. CVX services provide visibility and coordinate activities across a network of switches that are configured as CVX clients.

The **cvx** command enters CVX configuration mode. CVX configuration mode is not a group-change mode; ***running-config*** is changed immediately upon entering commands. Exiting CVX configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

The **no cvx** and **default cvx** commands restore all CVX server defaults by deleting all CVX configuration mode statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
cvx
no cvx
default cvx
```

Commands Available in CVX Configuration Mode

- **port (CVX)**
- **service openstack**
- **service vxlan**
- **shutdown (CVX)**
- **heartbeat-interval (CVX)**
- **heartbeat-timeout (CVX)**

Example

- These commands enter CVX-configuration mode and display the CVX configuration.

```
switch(config)#cvx
switch(config-cvx)#show active all

 cvx
   shutdown
   port 9979
   heartbeat-interval 20
   heartbeat-timeout 60
   no service vxlan
   service openstack
      shutdown
      name-resolution interval 21600
switch(config-cvx)#
```

heartbeat-interval (CVX)

The **heartbeat-interval** command configures the interval between heartbeat messages that the switch sends as a CVX server. Heartbeat messages are part of the keepalive mechanism between CVX and the CVX clients to which it connects.

The **no heartbeat-interval** and **default heartbeat-interval** commands restore the heartbeat interval to the default setting by removing the **heartbeat-interval** command from *running-config*.

Command Mode

CVX Configuration

Command Syntax

```
heartbeat-interval period
no heartbeat-interval
default heartbeat-interval
```

Parameters

- *period* Interval duration (seconds). Value ranges from 5 through 60. Default value is 20.

Related Commands

- **cvx** places the switch in CVX configuration mode.
- **heartbeat-timeout (CVX)** specifies CVX timeout interval.

Guidelines

Heartbeat messages flow independently in both directions between CVX and clients. When a client stops receiving heartbeat messages from the server within a specified period, the client assumes that the CVX server is no longer functioning.

Best practices dictate that CVX and its client applications configure identical heartbeat interval values.

Examples

- This command configures a CVX server heartbeat interval of 30 seconds:

```
switch(config)#cvx
switch(config-cvx)#heartbeat-interval 30
switch(config-cvx)#
```

heartbeat-interval (Management-CVX)

The **heartbeat-interval** command configures the interval between heartbeat messages that the switch sends as a CVX client. Heartbeat messages are part of the keepalive mechanism between the CVX client and the CVX server to which it connects.

The **no heartbeat-interval** and **default heartbeat-interval** commands revert the heartbeat interval to the default setting by removing the **heartbeat-interval** command from *running-config*.

Command Mode

Mgmt-CVX Configuration

Command Syntax

```
heartbeat-interval period
no heartbeat-interval
default heartbeat-interval
```

Parameters

- *period* Interval duration (seconds). Value ranges from 5 through 60. Default value is 20.

Guidelines

Heartbeat messages flow independently in both directions between CVX and clients. When the server stops receiving heartbeat messages from a client within a specified period, the server assumes that the device it is no longer functioning as a CVX client.

Best practices dictate that the CVX client's heartbeat interval value is identical to that of its CVX server.

Related Commands

- **management cvx** places the switch in Mgmt-CVX configuration mode.
- **heartbeat-timeout (Management-CVX)** specifies the CVX client timeout interval.

Examples

- These commands configure a CVX client heartbeat interval of 30 seconds:

```
switch(config)#management cvx
switch(config-mgmt-cvx)#heartbeat-interval 30
switch(config-mgmt-cvx)#
```

heartbeat-timeout (CVX)

The **heartbeat-timeout** command specifies the CVX timeout period. When a CVX server does not receive consecutive heartbeat messages from a CVX client within the heartbeat timeout period, the server discontinues providing CVX services to the client device. The default timeout period is 60 seconds.

The **no heartbeat-timeout** and **default heartbeat-timeout-timeout** commands restore the heartbeat timeout to the default setting by removing the **heartbeat-timeout** command from ***running-config***.

Command Mode

CVX Configuration

Command Syntax

heartbeat-timeout period
no heartbeat-timeout
default heartbeat-timeout

Related Commands

- **cvx** places the switch in CVX configuration mode.
- **heartbeat-interval (CVX)** specifies the CVX heartbeat interval.

Parameters

- *period* heartbeat timeout interval (seconds). Value ranges from ***15*** to ***10800***. Default value is 60.

Guidelines

Best practices dictate that CVX and its client applications configure identical heartbeat timeout values.

Examples

- These commands set the CVX timeout period to 90 seconds.

```
switch(config)#cvx
switch(config-cvx)#heartbeat-timeout 90
switch(config-cvx)#
```

heartbeat-timeout (Management-CVX)

The **heartbeat-timeout** command specifies the CVX client timeout period. When a CVX client does not receive consecutive heartbeat messages from a CVX server within the period specified by this command, the client assumes that its connection to CVX is disrupted. The default timeout period is 60 seconds.

The **no heartbeat-timeout** and **default heartbeat-timeout-timeout** commands restore the CVX client heartbeat timeout to the default setting by removing the **heartbeat-timeout** command from *running-config*.

Command Mode

Mgmt-CVX Configuration

Command Syntax

```
heartbeat-timeout period
no heartbeat-timeout
default heartbeat-timeout
```

Parameters

- *period* heartbeat timeout interval (seconds). Value ranges from ***15*** to ***10800***. Default value is 60.

Guidelines

Best practices dictate that the CVX client's heartbeat timeout value is identical to that of its CVX server.

Related Commands

- **management cvx** places the switch in Mgmt-cvx configuration mode.
- **heartbeat-interval (Management-CVX)** specifies the CVX client heartbeat interval.

Examples

- These commands set the CVX client timeout period to 90 seconds.

```
switch(config)#management cvx
switch(config-mgmt-cvx)#heartbeat-timeout 90
switch(config-mgmt-cvx)#
```

lldp run

The **lldp run** command enables LLDP on the Arista switch.

Command Mode

Global Configuration

Command Syntax

```
lldp run
no lldp run
default lldp run
```

Examples

- This command enables LLDP globally on the Arista switch.

```
switch(config)# lldp run
switch(config)#
```

- This command disables LLDP globally on the Arista switch.

```
switch(config)# no lldp run
switch(config)#
```

management cvx

The **management cvx** command places the switch in mgmt-CVX configuration mode to configure CVX client parameters.

Mgmt-CVX configuration mode is not a group-change mode; ***running-config*** is changed immediately upon entering commands. Exiting mgmt-CVX configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

The **no management cvx** and **default management cvx** commands delete all mgmt-CVX configuration mode statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
management cvx
no management cvx
default management cvx
exit
```

Commands Available in Mgmt-CVX Configuration Mode

- **heartbeat-interval (Management-CVX)**
- **heartbeat-timeout (Management-CVX)**
- **server host (Management-CVX)**
- **source-interface (Management-CVX)**
- **shutdown (Management-CVX)**

Example

- This command places the switch in mgmt-CVX configuration mode:

```
switch(config)#management cvx
switch(s1)(config-mgmt-cvx)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-cvx)#exit
switch(config)#
```

manager

The **manager** command configures the IP address of the OVSDB controller for the HSC service, allowing CVX to connect to the controller.

The **no manager** and **default manager** commands remove the HSC manager configuration from *running-config*.

Command Mode

CVX-HSC Configuration

Command Syntax

manager ip_address [port]

Parameters

ip_address **IP address of the HSC manager.**
port **connection port. Values range from 1 to 65535; default value is 6632.**

Related Commands

- **service hsc** places the switch in CVX-HSC configuration mode.

Example

- These commands point the HSC service to a controller at IP address 192.168.2.5 using the default port 6632.

```
switch(config)#cvx
switch(config-cvx)#service hsc
switch(config-cvx-hsc)#manager 192.163.2.5
switch(config-cvx-hsc)#
```

name-resolution force (CVX-OpenStack)

The **name-resolution force** command initiates an OpenStack controller function that communicates with the OpenStack Keystone and Nova services to update names of VMs and tenants mapped by the local OpenStack instance.

The OpenStack controller accesses the Keystone and Nova services in response to various triggering events (such as the creation of a new tenant, network or VM), and also at a regular interval configured by the **name-resolution interval (CVX-OpenStack)** command (default interval 6 hours). The **name-resolution force** command is used to force an immediate update without waiting for a triggering event.

Command Mode

CVX-OpenStack Configuration

Command Syntax

```
name-resolution force
```

Related Commands

- **service openstack** places the switch in CVX-OpenStack configuration mode.
- **name-resolution interval (CVX-OpenStack)** sets the interval for automatic Keystone updates.

Example

- These commands update the OpenStack instance immediately with data from the Keystone service.

```
switch(config)#cvx
switch(config-cvx)#service openstack
switch(config-cvx-openstack)#name-resolution force
switch(config-cvx-openstack)#
```

name-resolution interval (CVX-OpenStack)

The **name-resolution interval** command specifies the period between consecutive requests that the OpenStack controller sends to the Keystone service for VM and tenant name updates. Keystone is OpenStack's authentication and authorization service.

The default period is 21600 seconds (6 hours).

The **name-resolution force (CVX-OpenStack)** command performs an immediate update, as opposed to waiting for the periodic update.

Command Mode

CVX-OpenStack Configuration

Command Syntax

name-resolution interval period

Parameters

- *period* Keystone identity service polling interval (seconds).

Related Commands

- **service openstack** places the switch in CVX-OpenStack configuration mode.

Example

- These commands set the name resolution interval period at five hours.

```
switch(config)#cvx
switch(config-cvx)#service openstack
switch(config-cvx-openstack)#name-resolution interval 18000
switch(config-cvx-openstack)#
```

ovsdb-shutdown

The **ovsdb-shutdown** command shuts down the OVSDB server.

The **no ovsdb-shutdown** and **default ovsdb-shutdown** commands enable the OVSDB server by removing the **ovsdb-shutdown** command from *running-config*.

Command Mode

CVX-HSC Configuration

Command Syntax

```
ovsdb-shutdown
no ovsdb-shutdown
default ovsdb-shutdown
```

Related Commands

- **service hsc** places the switch in CVX-HSC configuration mode.

Example

- These commands shut down the OVSDB server used by the HSC service.

```
switch(config)#cvx
switch(config-cvx)#service hsc
switch(config-cvx-hsc)#ovsdb-shutdown
switch(config-cvx-hsc)#
```

port (CVX)

The **port** command specifies the TCP port number the CVX server listens on. The default port number is 9979.

The **no port** and **default port** commands restore the default port number by removing the **port** statement from *running-config*.

Command Mode

CVX Configuration

Command Syntax

```
port port_number
no port
default port
```

Parameters

- *port_number* TCP port number. Value ranges from *1* to *65535*.

Related Commands

- **cvx** places the switch in CVX configuration mode.

Example

- These commands configure 9500 as the CVX server port.

```
switch#config
switch(config)#cvx
switch(config-cvx)#port 9500
switch(config-cvx)#
```

- These commands restore the default port (9979) as the CVX server port.

```
switch(config-cvx)#no port
switch(config-cvx)#
```

resync-period

The **resync-period** command configures the grace period for completion of synchronization between the VXLAN control service and clients after a CVX restart. Arista recommends leaving the grace period set to its default of 300 seconds.

The **no resync-period** command disables VXLAN control service graceful restart. The **default resync-period** command resets the grace period to its default of 300 seconds.

Command Mode

CVX-VXLAN Configuration

Command Syntax

```
resync-period <seconds>
no resync-period
default resync-period
```

Parameters

- *seconds* synchronization grace period in seconds. Values range from 30 to 4800; default is 300.

Examples

- These commands reset the VXLAN control service synchronization grace period to 300 seconds.

```
switch(config)#cvx
switch(config-cvx)#service vxlan
switch(config-cvx-vxlan)#default resync-period
switch(config-cvx-vxlan)#
```

server host (Management-CVX)

The **server host** command configures the IP address or host name of the CVX server to which the CVX client device connects. The configuration of this address is required for the switch to function as a CVX client. By default, no CVX host address is specified.

The **no server host** and **default server host** commands remove the CVX host address assignment by removing the **server host** statement from *running-config*.

Command Mode

Mgmt-CVX Configuration

Command Syntax

```
server host host
no server host
default server host
```

Parameters

- *host* IPv4 address (in dotted decimal notation) or FQDN host name of the CVX server.

Related Commands

- **management cvx** places the switch in Mgmt-CVX configuration mode.

Examples

- This command specifies 10.1.1.14 as the address of the server to which the CVX client connects.

```
switch(config)#management cvx
switch(config-mgmt-cvx)#server host 10.1.1.14
switch(config-mgmt-cvx)#
```

service hsc

The **service hsc** command enters CVX-HSC configuration mode where the hardware switch controller (HSC) service is enabled and configured.

CVX-HSC configuration mode is not a group change mode; *running-config* is changed immediately upon entering commands. Exiting CVX-HSC configuration mode does not affect *running-config*. The **exit** command returns the switch to global configuration mode.

Command Mode

CVX Configuration

Command Syntax

```
service hsc
```

Commands Available in CVX-HSC Configuration Mode

- **manager**
- **ovsdb-shutdown**
- **shutdown (CVX-HSC)**
- **vtep (CVX-HSC)**

Related Commands

- **cvx** places the switch in CVX configuration mode.

Example

- These commands enter CVX-HSC configuration mode.

```
switch(config)#cvx
switch(config-cvx)#service hsc
switch(config-cvx-hsc)#
```

service openstack

The **service openstack** command places the switch in CVX-OpenStack configuration mode.

In order to integrate Arista switches into an OpenStack managed cloud network, OpenStack needs to interact with CVX to configure and maintain VLANs on appropriate physical switch ports that connect to hosts where the VMs reside.

CVX-OpenStack configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting CVX-OpenStack configuration mode does not affect ***running-config***. The **exit** command returns the switch to global configuration mode.

Command Mode

CVX Configuration

Command Syntax

```
service openstack
```

Commands Available in CVX-OpenStack Configuration Mode

- **name-resolution force (CVX-OpenStack)**
- **name-resolution interval (CVX-OpenStack)**
- **shutdown (CVX-OpenStack)**

Related Commands

- **cvx** places the switch in CVX configuration mode.

Example

- These commands places the switch in CVX-OpenStack configuration mode.

```
switch(config)#cvx
switch(config-cvx)#service openstack
switch(config-cvx-openstack)#
```

service vxlan

The **service vxlan** command enters CVX-VXLAN configuration mode where the VXLAN control service is enabled and configured.

CVX-VXLAN configuration mode is not a group change mode; *running-config* is changed immediately upon entering commands. Exiting CVX-VXLAN configuration mode does not affect *running-config*. The **exit** command returns the switch to global configuration mode.

Command Mode

CVX Configuration

Command Syntax

```
service vxlan
```

Commands Available in CVX-VXLAN Configuration Mode

- **resync-period**
- **shutdown (CVX-VXLAN)**
- **vtep (CVX-VXLAN)**

Related Commands

- **cvx** places the switch in CVX configuration mode.

Example

- These commands enter CVX-VXLAN configuration mode.

```
switch(config)#cvx
switch(config-cvx)#service vxlan
switch(config-cvx-vxlan)#
```

show cvx

The **show cvx** command displays the enable status and current configuration of CVX.

Command Mode

EXEC

Command Syntax

show cvx

Example

- This command displays status and configuration of CVX.

```
switch(config)#cvx

 cvx
   no shutdown
   heartbeat-interval 30
   heartbeat-timeout 90
switch(config-cvx)#dis
switch>show cvx
CVX Server
   Status: Enabled
   UUID: 75ce27ce-cc04-11e4-a404-233646319a2c
   Heartbeat interval: 30.0
   Heartbeat timeout: 90.0
switch>
```

show network physical-topology

The **show network physical-topology** command displays the network topology discovered through CVX.

Command Mode

EXEC

Command Syntax

```
show network physical-topology hosts|neighbors
```

Parameters

- **hosts** Displays all hosts visible in the topology.
- **neighbors** Displays all connections in the network topology. Table is sorted by host name, and can be optionally filtered by host.

Example

- This command displays all visible hosts.

```
switch#show network physical-topology hosts
Unique Id            Hostname
-------------------- ------------------------------
001c.7385.be69       cvx287.sjc.aristanetworks.com
0000.6401.0000       cvc1
0000.6402.0000       cvc2
0000.6403.0000       cvc3
0000.6404.0000       cvc4
bcf6.85bd.8050       dsj14-rack14-tor1
```

- This command displays all connections in the topology.

```
switch#show network physical-topology neighbors
cvx287.sjc.aristanetworks.com
Interface Neighbor Intf Neighbor Host
----------------- ----------------- ----------------------
Ethernet1         Ethernet7 cvc4
Ethernet2         Ethernet7 cvc2
Ethernet9         Ethernet7 cvc1
Ethernet10        Ethernet7 cvc3
Management1 27    dsj14-rack14-tor1

                <-------OUTPUT OMITTED FROM EXAMPLE-------->

dsj14-rack14-tor1
Interface Neighbor Intf Neighbor Host
----------------- ----------------- ----------------------
27 Management1    cvx287.sjc.aristanetwork
```

shutdown (CVX)

The **shutdown** command, in cvx mode, disables or enables the switch as a CVX server. By default, CVX is disabled on the switch.

The **no shutdown** command enables the switch as a CVX server. The **shutdown** and **default shutdown** commands disable the switch as a CVX server by removing the **no shutdown** command from *running-config*.

Important! Be sure to de-configure or shut down all CVX client services before disabling CVX; failure to do so may result in CVX client services continuing to run after CVX has been disabled.

Command Mode

CVX Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **cvx** places the switch in CVX configuration mode.

Example

- These commands enable the switch as a CVX server.

```
switch#config
switch(config)#cvx
switch(config-cvx)#no shutdown
switch(config-cvx)#
```

- This command disables CVX on the switch.

```
switch(config-cvx)#shutdown
switch(config-cvx)#
```

shutdown (CVX-HSC)

The **shutdown** command, in CVX-HSC configuration mode, disables or enables the CVX hardware switch controller (HSC) service on the switch. HSC is disabled by default.

When a CVX server enables HSC, its clients (hardware VTEPs) are able to share state to establish VXLAN tunnels without the need for a multicast control plane. Configuration is also required on the client switches.

The **no shutdown** command enables the HSC service; the **shutdown** and **default shutdown** commands disable the HSC service.

Command Mode

CVX-VXLAN Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **service hsc** places the switch in CVX-HSC configuration mode.

Example

- These commands enable the HSC service.

```
switch(config)#cvx
switch(config-cvx)#service hsx
switch(config-cvx-hsc)#no shutdown
switch(config-cvx-hsc)#
```

- These commands disable the HSC service.

```
switch(config)#cvx
switch(config-cvx)#service hsx
switch(config-cvx-hsc)#shutdown
switch(config-cvx-hsc)#
```

shutdown (CVX-OpenStack)

The **shutdown** command, in cvx-openstack configuration mode, disables or enables CVX-OpenStack on the switch. CVX-OpenStack is disabled by default.

When a CVX server enables OpenStack services, its clients are accessible to the OpenStack network controller (Neutron). Integrating Arista switches into an OpenStack-managed cloud network requires OpenStack to interact with CVX to configure and maintain VLANs on appropriate physical switch ports that connect to the hosts where the VMs reside.

The **no shutdown** command enables CVX-OpenStack. The **shutdown** and **default shutdown** commands disable CVX-OpenStack by removing the corresponding **no shutdown** command from *running-config*.

Command Mode

CVX-OpenStack Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **service openstack** places the switch in CVX-OpenStack configuration mode.

Example

- These commands enable CVX-OpenStack.

```
switch(config)#cvx
switch(config-cvx)#service openstack
switch(config-cvx-openstack)#no shutdown
switch(config-cvx-openstack)#
```

- These commands disable CVX-OpenStack.

```
switch(config-cvx-openstack)#
switch(config-cvx-openstack)#shutdown
switch(config-cvx-openstack)#
```

shutdown (CVX-VXLAN)

The **shutdown** command, in CVX-VXLAN configuration mode, disables or enables the CVX VXLAN control service on the switch. VXLAN control service is disabled by default.

When a CVX server enables VXLAN control service, its clients (hardware VTEPs) are able to share state to establish VXLAN tunnels without the need for a multicast control plane. Configuration is also required on the client switches.

The **no shutdown** command enables the VXLAN control service. The **shutdown** and **default shutdown** commands disable the VXLAN control service.

Command Mode

CVX-VXLAN Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **service vxlan** places the switch in CVX-VXLAN configuration mode.

Example

- These commands enable VXLAN control service.

```
switch(config)#cvx
switch(config-cvx)#service vxlan
switch(config-cvx-vxlan)#no shutdown
switch(config-cvx-vxlan)#
```

- These commands disable VXLAN control service

```
switch(config)#cvx
switch(config-cvx)#service vxlan
switch(config-cvx-vxlan)#shutdown
switch(config-cvx-vxlan)#
```

shutdown (Management-CVX)

The **shutdown** command, in mgmt-cvx mode, disables or enables CVX client services on the switch. CVX services are disabled by default.

The **no shutdown** command enables CVX client services. The **shutdown** and **default shutdown** commands disable CVX client services by removing the corresponding **no shutdown** command from *running-config*.

Command Mode

Mgmt-CVX Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Related Commands

- **management cvx** places the switch in Mgmt-cvx configuration mode.

Example

- These commands enable CVX client services.

```
switch(config)#management cvx
switch(config-mgmt-cvx)#no shutdown
switch(config-mgmt-cvx)#
```

- This command disables CVX client services.

```
switch(config-mgmt-cvx)#shutdown
switch(config-mgmt-cvx)#
```

source-interface (Management-CVX)

The **source-interface** command specifies the interface from where the IPv4 address is derived for use as the source for outbound CVX packets that the switch sends as a CVX client. There is no default source interface assignment.

The **no source-interface** and **default source-interface** commands remove the source interface assignment for the CVX client by deleting the **source-interface** statement from *running-config*.

Command Mode

Mgmt-CVX Configuration

Command Syntax

```
source-interface INT_NAME
no source-interface
default source-interface
```

Parameters

- *INT_NAME* Interface type and number. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Related Commands

- **management cvx** places the switch in Mgmt-CVX configuration mode.

Example

- These commands configure the CVX client to use the IP address 10.24.24.1 as the source address for its outbound packets.

```
switch#config
switch(config)#interface loopback 5
switch(config-if-Lo5)#ip address 10.24.24.1/24
switch(config-if-Lo5)#exit
switch(config)#management cvx
switch(config-mgmt-cvx)#source-interface loopback 5
switch(config-mgmt-cvx)#
```

vtep (CVX-HSC)

The HSC service sends flood lists to each VTEP through CVX. Some controllers (such as VMware NSX's Service Nodes) implement replication nodes for head-end replication of unknown packets. For these controllers, BUM packets should be sent to a single replication node (send-to-any replication), and the flood list sent by the HSC service is a list of replication nodes. Other controllers (such as Nuage VSP) require each VTEP to perform its own head-end replication. For these, BUM packets should be sent to every known VTEP, and the flood list sent by the HSC service is the list of VTEPs.

The default behavior is to use a send-to-any replication list of VTEPs. If the required behavior is send-to-all replication of, use the **all** option of the **vtep** command in CVX-HSC configuration mode.

Command Mode

CVX-HSC Configuration

Command Syntax

```
vtep flood list type all|any
no vtep flood list type
default vtep flood list type
```

Parameters

- all send-to-all replication; flood list is the list of VTEPs.
- any send-to-any replication; flood list is a list of replication nodes. This is the default setting.

Related Commands

- **service hsc** places the switch in CVX-HSC configuration mode.

Example

- These commands configure the HSC to use send-to-all replication.

```
switch(config)#cvx
switch(config-cvx)#service hsc
switch(config-cvx-hsc)#vtep flood list type all
switch(config-cvx-hsc)#
```

vtep (CVX-VXLAN)

The OVSDB management protocol includes provisions for control-plane MAC learning, which allows MAC addresses to be distributed among VTEPs without using the data plane. Some controllers (such as VMware NSX) take advantage of this facility; others (such as Nuage VSP) do not. By default, CVX uses control-plane MAC learning.

To switch to data plane MAC learning, use the **vtep** command in CVX-VXLAN configuration mode, as shown below.

Command Mode

CVX-VXLAN Configuration

Command Syntax

```
vtep mac-learning control-plane|data-plane
```

Related Commands

- **service vxlan** places the switch in CVX-VXLAN configuration mode.

Example

- These commands configure CVX to use data-plane MAC address learning.

```
switch(config)#cvx
switch(config-cvx)#service vxlan
switch(config-cvx-vxlan)#vtep mac-learning data-plane
switch(config-cvx)#
```

Related Topics

- "CVX Overview" on page 4
- "CVX Services" on page 5
- "Deploying CVX" on page 6
- "CVX Configuration" on page 14
- "CVX Secure out-of-band Connection" on page 20
- "CVX High Availability" on page 24